

Nattan MAISONNEUVE–FLEURY
BTS SIO 1
Groupe B

04/09/2025

Compte Rendu - Premier pas en Java

Sommaire :

Introduction :	1
Notre premier code :	1
étapes 1 - Créer un environnement de travail:	1
étapes 2 - Le code :	2
Comprendre nos premières ligne :	3
Conclusion :	5

Introduction :

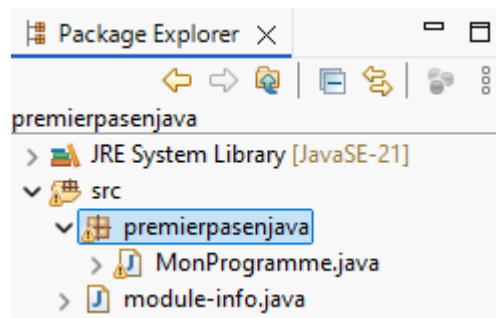
Dans ce TP nous allons voir nos premier ligne en langage Java. C'est un logiciel utilisé dans les applications web, applications Android, applications de bureau... Java est très populaire grâce à sa portabilité (*il fonctionne sur toutes les plateformes grâce à sa machine virtuelle*), grâce à sa grande communauté qui propose des bibliothèques. Enfin, le langage offre de bonnes performances et est un langage évolutif ce qui le rend adapté aux applications de grande envergure.

Notre premier code :

Partons du fait que Eclipse (Logiciel ou l'on va stocker notre espace de travail JS et écrire du code) ainsi que Java sont déjà téléchargés.

étapes 1 - Créer un environnement de travail:

1. Lancez l'application eclipse.
2. En haut à gauche appuyer sur **"File"** -> **"Java Project"**.
3. Une fenêtre s'ouvre. Dans 'project name' modifier le nom par le nom que vous voulez donner à votre projet, par exemple **premierpasenjava** puis appuyez sur **"Finish"**.
4. Sur la gauche faite clique droit sur le nom de votre projet



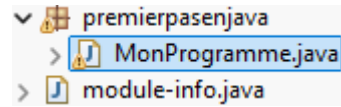
Appuyer sur **"New"** -> **"Class"**

5. Une nouvelle fenêtre apparaît. Modifier la case 'Name' par un nouveau nom **MonProgramme**, ensuite cochez la case 'public static void main(Stringg[] args)'. Et enfin, appuyez sur **"Finish"**.

étapes 2 - Le code :

Notre environnement de travail est créé, nous allons pouvoir commencer à coder.

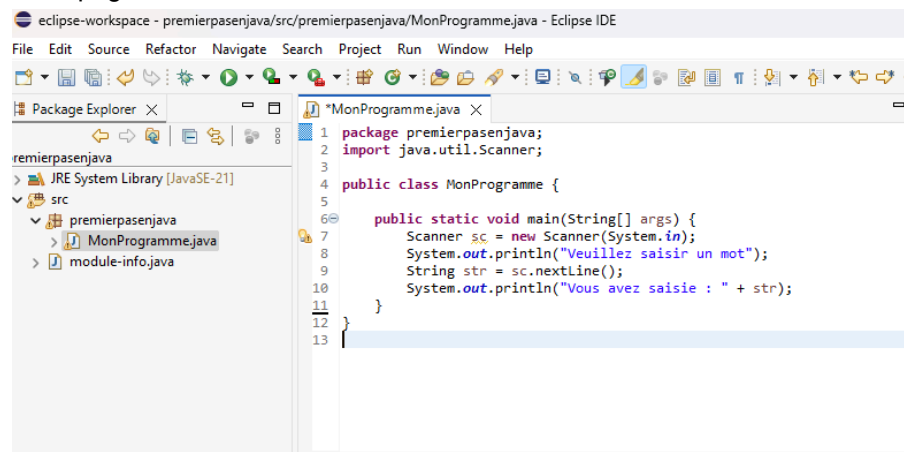
1. Sur la gauche appuyer sur le fichier MonProgramme.java



2. Ensuite, dans la page qui vient de s'ouvrir mettez ce code :

```
package premierpasenjava;
import java.util.Scanner;
public class MonProgramme {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Veuillez saisir un mot");
        String str = sc.nextLine();
        System.out.println("Vous avez saisie : " + str);
    }
}
```

Votre page devrait ressembler à ceci :



3. Maintenant appuyer sur ce bouton pour dire au programme de se lancer :



4. Si vous n'avez pas fait d'erreur, dans la console en bas, vous devriez avoir écrit

- Veuillez saisir un mot

Écrivez donc un mot, exemple bonjour.

Le code devrez vous répondre

- Vous avez saisie : bonjour

Si vous avez une erreur, c'est sûrement que vous avez faits une fautes de frappe sur un mot, une virgule, une parenthèse...

Comprendre nos premières ligne :

Dans ce code, nous avons plusieurs lignes de code et nous allons essayer de les comprendre. Avant de commencer il faut bien savoir qu'à la fin de chaque ligne, il y a un point virgule pour nous dire 'fin de l'instruction'

1 - **package** premierpasenjava;

Cette ligne est sûrement la plus simple. Le premier mot, **package** veut dire 'ouvrir le dossier', et 'premierpasenjava' est le nom d'un dossier que nous avons créé. A la suite, cela nous donne : 'ouvrir le dossier premierpasenjava' (dans le dossier, il faut bien sur avoir un fichier .java). Si nous avons créé un dossier 'test' alors nous aurions écrit **package** test.

2 - **import** java.util.Scanner;

La ligne '**import** java.util.Scanner;' en Java est une instruction qui permet d'importer la classe **Scanner** de la bibliothèque standard de Java, située dans le package `java.util`.

Pour vraiment la comprendre, décortiquons la en 3

import - un mot clé pour indiquer que l'on veut utiliser une classe, une interface ou un autre type d'élément provenant d'un autre package dans notre code. Sans cette ligne on ne pourrait pas utiliser la classe **Scanner** ou n'importe quelle autre classe si elle est située dans un package différent.

java.util - C'est le nom du package dans lequel se trouve la classe `Scanner`. Le package `java.util` contient de nombreuses classes utiles, comme des structures de données (Listes, Set, Map), des utilitaires pour la gestion du temps et des dates, et d'autres outils comme `Scanner`.

.Scanner - C'est la classe qui permet de lire des entrées de l'utilisateur depuis le clavier, des fichiers, ou d'autres flux d'entrée. Elle est très couramment utilisée pour prendre des données comme des chaînes de texte, des nombres, etc., dans les applications console.

3 - **public class** MonProgramme {

public - signifie que la classe peut être accédé depuis n'importe où.

class - est le mot-clé pour déclarer une classe.

MonProgramme - nom de la classe qui correspond au nom du fichier `MonProgramme.java`.

{ - marque le début du bloc de code de la classe.

4 - **public static void** main(String[] args) {

public – La méthode peut être accédée depuis n'importe où, y compris en dehors de la classe.

static – La méthode appartient à la classe elle-même, pas à une instance de la classe.

void – La méthode ne retourne aucune valeur.

main – Nom de la méthode, c'est le point d'entrée du programme Java.

String[] args – Un tableau de chaînes de caractères qui permet de recevoir des arguments passés au programme lors de son lancement.

{ – Marque le début du bloc de code de la méthode main.

5 - Scanner **sc** = **new** Scanner(System.**in**);

Scanner – Nom de la classe qui permet de lire des entrées, comme du texte ou des nombres.

sc – Nom de l'objet Scanner. C'est une référence qui permet d'utiliser l'instance de la classe Scanner pour lire des données.

= – L'opérateur d'affectation, qui permet d'assigner une valeur à la variable **sc**.

new – Utilisé pour créer une nouvelle instance (un objet) de la classe Scanner.

Scanner (System.in) – Crée un objet Scanner qui lit l'entrée de l'utilisateur depuis System.**in**, c'est-à-dire le clavier.

6 - System.**out**.println("Veuillez saisir un mot");

System – Représente la classe System qui contient des outils pour interagir avec le système, comme l'entrée/sortie (I/O).

out – Un objet de la classe PrintStream qui permet d'afficher des messages à l'écran (sortie standard).

println – Une méthode de PrintStream qui affiche un message et passe à la ligne suivante après l'affichage.

("Veuillez saisir un mot") – Le texte à afficher. Ici, c'est le message **"Veuillez saisir un mot"** qui sera affiché dans la console.

7 - String **str** = **sc.nextLine()**;

String – Type de la variable, ici une chaîne de caractères (texte).

str – Nom de la variable qui va contenir la chaîne de caractères lue depuis l'entrée utilisateur.

= – L'opérateur d'affectation, qui permet d'assigner une valeur à la variable **str**.

sc – Objet de type Scanner, utilisé pour lire les données de l'utilisateur.

nextLine() – Méthode de l'objet Scanner qui permet de lire une ligne entière de texte (jusqu'à ce que l'utilisateur appuie sur "Entrée").

8 - System.**out**.println("Vous avez saisi : " + **str**);

System – La classe System, qui fait partie de la bibliothèque standard Java, permet d'interagir avec l'entrée et la sortie du système (I/O).

out – Un objet PrintStream de la classe System, utilisé pour afficher des messages à l'écran (sortie standard).

println – Une méthode de PrintStream qui permet d'afficher un message et de passer à la ligne suivante après l'affichage.

"Vous avez saisi : " – Texte à afficher avant la valeur de la variable **str**. C'est une chaîne de caractères qui sera imprimée telle quelle.

+ – L'opérateur de **concaténation** qui permet de combiner (concaténer) deux chaînes de caractères.

str – La variable contenant la chaîne de caractères que l'utilisateur a saisie.

9 - }

Ces accolades sont utilisées pour fermer un bloc.

Conclusion :

Aujourd'hui, nous avons pu voir nos premières lignes de Java mais surtout apprendre à quoi ce langage sert. Grâce à cela nous allons pouvoir rentrer pleinement dans le codage Java et le développement d'applications.